AD-A063 936 CONSTRUCTION ENGINEERING RESEARCH LAB (ARMY) CHAMPAI--ETC F/G 9/2
RECOMMENDED INTERFACE STANDARDS FOR AN ARMY STANDARD ENERGY MON--ETC(U)
DEC 78 J L HALL

UNCLASSIFIED CERL-TR-E-140 NL

| OF |

AD
A063936

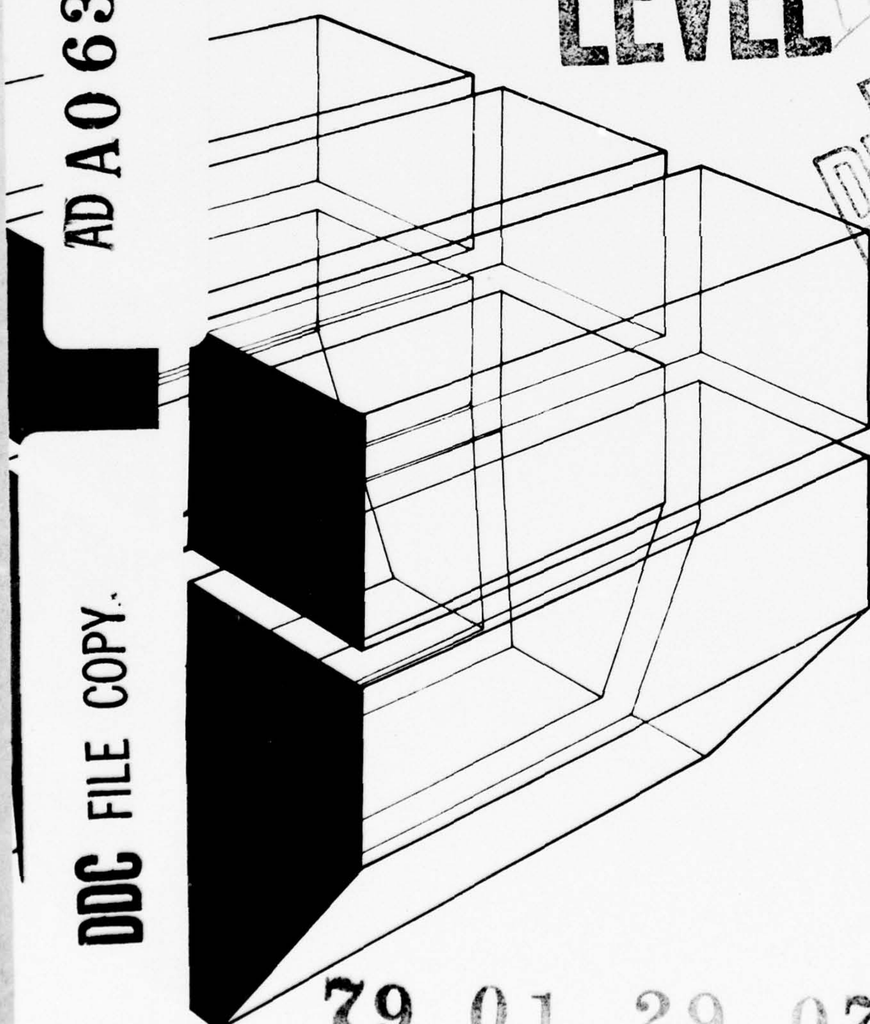
END
DATE
FILMED
3-79
DDC

construction engineering research laboratory

12

TECHNICAL REPORT E-140
December 1978

RECOMMENDED INTERFACE STANDARDS FOR AN ARMY STANDARD ENERGY MONITORING AND CONTROL SYSTEM



by
J. Hall

CERL

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) CERL-TR-E-140	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) RECOMMENDED INTERFACE STANDARDS FOR AN ARMY STANDARD ENERGY MONITORING AND CONTROL SYSTEM.		5. TYPE OF REPORT & PERIOD COVERED (9) FINAL rept,
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Hall (10) James L. Hall		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. ARMY CONSTRUCTION ENGINEERING RESEARCH LABORATORY P.O. Box 4005, Champaign, IL 61820	(16)	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A762721AT41 T6-010 (17)
11. CONTROLLING OFFICE NAME AND ADDRESS	(11)	12. REPORT DATE Dec[redacted]78
		13. NUMBER OF PAGES 34 (12) 37 p.
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Copies are obtainable from National Technical Information Service
Springfield, VA 22151

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

interfacing standards
energy monitoring and control system
serial data interfaces

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report evaluated existing interfacing standards for use with an Army Standard Energy Monitoring and Control System (EMCS). The interfacing requirements for the Army Standard EMCS and the criteria for making the required evaluations are presented. Descriptions of the existing standards which were found to be candidates for evaluation are presented, along with the results of the evaluations. (over)

(1)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE 405 279 UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued.

This report recommends that for the Army Standard EMCS, a concept similar to IEEE standard 583 be used for parallel data interface and that the EIA R232-C standard be used for serial data interfaces.

FOREWORD

This work was performed for the Directorate of Military Construction, Office of the Chief of Engineers (OCE), under Project 4A762731AT41, "Design, Construction, Operation and Maintenance, and Technology for Military Facilities"; Task T6, "Energy Systems"; Work Unit 010, "Fixed Facilities Energy Control Systems." Mr. James Walton was the OCE Technical Monitor.

This study was performed by the Energy and Power Division (EP), U.S. Army Construction Engineering Research Laboratory (CERL). Mr. R. G. Donaghy is Chief of EP.

COL J. E. Hays is Commander and Director of CERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

FIGURES

TABLES

RECOMMENDED INTERFACE STANDARDS FOR AN
ARMY STANDARD ENERGY MONITORING AND CONTROL SYSTEM

1 INTRODUCTION

Background

The feasibility of a standard energy monitoring and control system (EMCS) for the Army has been established.[1] The structure of the proposed system is a distributed system of local intelligent building controllers in communication with a central computer. For such a system to operate, there must be a method for passing information between the modules. Standardization of this method would be desirable, since this would provide maximum flexibility in system hardware, and therefore in system hardware suppliers. Individual system elements could be defined by function and interfacing requirements. If the interfacing requirements are standardized, elements of the system need only be defined by function.

Objective

The objective of this work is to evaluate available interfacing and data communications standards and to recommend the most efficient standards for the individual interfacing requirements based on the defined criteria for a standard, nonproprietary Army EMCS.

Approach

Data flow requirements for each data transmission path of the Army standard EMCS were determined, taking into account such physical restraints as distance and available communication system hardware. Based on these requirements, basic criteria for evaluating the standard communications methods were developed. Existing standards that meet these criteria were identified and examined for applicability to the Army standard EMCS to determine the most efficient standards for meeting interfacing requirements.

---

[1] D. Eng and K. H. Wu, *A Study of the Technical Feasibility of Developing a Standardized Energy Control System Specifically for Army Facilities*, Interim Report E-117/ADA044455 (U.S. Army Construction Engineering Research Laboratory [CERL], August 1977).

Outline of Report

Chapter 2 of this report briefly describes the proposed EMCS, discusses the requirements of the individual data communications interfaces, and establishes the criteria for evaluating the interfacing standards. Chapters 3 and 4 describe candidate serial data communications standards and parallel data communications bus standards, respectively. Chapter 5 provides an evaluation of the interfacing standards, and Chapter 6 presents conclusions. Chapter 7 outlines recommended data communications standards. The appendix provides further notes on the design of an acceptable parallel data communication bus.

Mode of Technology Transfer

Results of this work will be incorporated into a draft guide specification for a nonproprietary, Army standard EMCS.

2 CRITERIA FOR EVALUATION OF INTERFACING STANDARDS

Description of Proposed EMCS

The proposed Army standard EMCS was designed to overcome the following major problems associated with available industry-supplied systems.

1. Without major engineering changes, industry-supplied systems are compatible only with systems of the same manufacturer. No industry-wide standards exist.

2. Most systems require a large central computer for system control. A component failure in this element causes system failure.

3. Many systems are designed for single-building applications and not for the particular large-scale application required for performing energy control of a large, multi-building military installation.

The EMCS proposed to overcome these problems (see Figure 1) is designed to be as modular as possible, thus facilitating the capability for future system expansion. The intelligence required to perform energy monitoring and control is distributed as much as possible. The actual building energy control is performed at the building location by a microprocessor-based Field Interface Device (FID). This choice provides the capability for both isolated building energy control, and for connecting the FID's Unit (CCU) via appropriate communications channels to a central control capable of controlling installation-wide energy usage. Since the intelligence providing energy control is distributed, individual component failure does not fatally impact the total system operation. For instance, CCU failure will not cause total loss of system operation, since the FID's will still be capable of providing most of the energy control; only the installation-wide control algorithm execution will be lost.

Data communication between the CCU and the FID's is handled by the communications modules and is performed in serial fashion over twisted pair cables or the existing telephone systems, whichever is appropriate.

Interface Requirements

As shown in Figure 1, there are three definite areas of data communications. The first is the data flow between sensors and actuators and the Multiplex (MUX) panel. Data flow through this interface should be implemented with a straightforward parallel data bus to provide simple access to the building operating parameters by the FID. This should

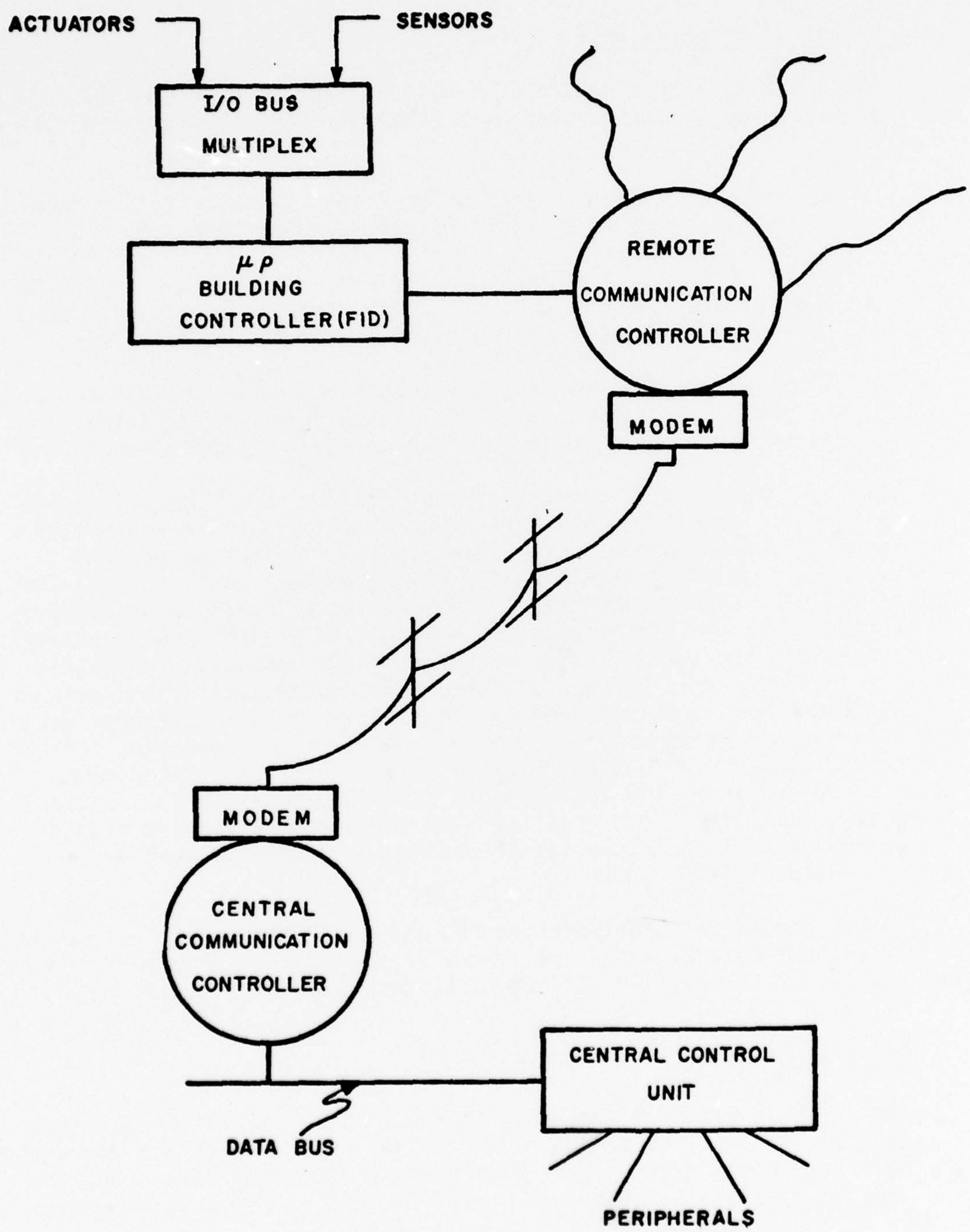


Figure 1. Proposed EMCS architecture.

minimize the amount of microprocessor software necessary to process the sensor data, control the actuators, and minimize the amount of interfacing hardware.

The second area of data communication is the serial data flow to and from the CCU via the communication controllers. Typically, the remote communications controller will be located to serve a group of FID's via serial data communications. Since this distance will probably be short, the serial data paths can be implemented with differential line drivers/receivers. However, for the longer transmission paths between the communications controllers, it will be necessary to implement serial data communications via the existing telephone system, using modulator/demodulator (modem) hardware and Data Access Arrangements (DAA). This data link is the weakest in the system, due to the inherent data dropouts, crosstalk, and bandwidth limitations incurred with the telephone system. It is necessary that all data transmitted over this link be appropriately coded to provide automatic error detection and correction by the communications controllers.

The third area of data communication is the parallel data communications interface between the CCU and the central communications controller (CCC). Data passed through this parallel data interface consists of operations parameters and setpoint information from all the buildings, and base-wide control information from the CCU to selected FID's. This interface must provide rapid access to the CCC to minimize the CCU waiting time.

Criteria for Evaluation of Interface Commands

Considering the requirements of the three interfaces, the following criteria were used to evaluate existing standards:

1. For the parallel data flow from sensors to the FID's and back to the actuators:

   a. Simplicity of bus structure: the number of lines required to implement the standard must be less than 32.

   b. Ease of operations, determined by the amount of hardware and software necessary to provide data flow over the interface.

2. For the serial data flow between the FID and the remote communications controller, and for the serial data flow between the remote and central communications controllers:

a. Implementation of the major hardware functions on a single integrated circuit.

b. Provisions for automatic error detection and recovery.

c. Capability for simultaneous bidirectional data transmission.

3. For the parallel data flow between the CCC and the CCU: absolute minimum of software control from the CCU required to maintain efficient interface operation.

Additional elements contained in all the required standards are (1) standard connector types and pin definitions, and (2) standard function descriptions or code characters.

3 SERIAL DATA COMMUNICATIONS STANDARDS

Selection of Standards

The selection of candidate serial data communication standards was complicated by the many standards or protocols available. Two of the standards, Electronics Industry Association (EIA) RS232-C,[2] as updated by RS422 and RS423, and IBM Synchronous Data Link Control (SDLC)[3] represent the most advanced forms of asynchronous and synchronous standards available and were thought to be able to best meet the system requirements. These two standards were therefore evaluated against the criteria given in Chapter 2.

IBM Synchronous Data Link Control Procedure (SDLC)

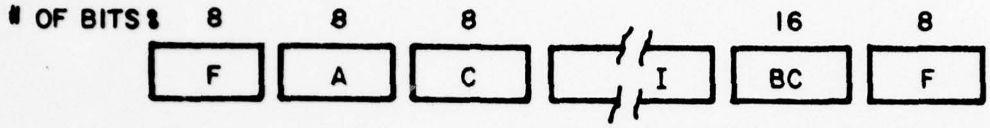
SDLC is a data link control for serial bit synchronous transmission between buffered stations on a data transmission link using centralized control. The data transmission link may be customer-owned, -leased, or -switched facilities connected in half-duplex, full-duplex, or logs configurations. SDLC provides a means by which information can be exchanged between an outlying station (called a secondary station) and one central station (called the primary station). The primary station initiates or authorizes all transmissions regardless of the direction of information flow, by issuing commands to secondary stations. SDLC includes a method for detecting transmissions errors and a method for correcting these errors by retransmission.

All transmissions on the link have a specific format called a frame. A frame is delimited by a unique bit sequence called a flag at the beginning and end of each transmission block. Between the flags, a frame contains a station address field, a control field, an information field, and a block check field (see Figure 2). Theoretically there is no limit to the length of the information field; however, there is a practical limit based on the limitations of the error-checking algorithm, the statistics of errors on the communications channel, and the buffering capacity of the individual stations.

---

[2] *EIA Standard -- Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*, (Electronic Industries Association, August 1969).

[3] R. A. Donnan and J. R. Kersey, "Synchronous Data Link Control: A Perspective," *IBM Systems Journal*, Vol 13, No. 2 (1974) pp 140-162.



F: FLAG SEQUENCE FIELD

A: STATION ADDRESS FIELD

C: CONTROL FIELD

I: INFORMATION (DATA) FIELD

BC: BLOCK CHECK FIELD

Figure 2. SDLC frame format.

A single large-scale integrated circuit (LSI) that provides processor transparency of the SDLC protocol has recently been made available by Signetics, Inc., Sunnyvale, CA.

The SDLC protocol does not include a method for delimiting records, messages, or any other units of data other than the information fields within frames. Furthermore, device addressing, device control, and device status are all beyond the scope of SDLC. Under SDLC, there is no way for secondary stations to inform the primary station of alarm conditions unless they are polled by the primary station.

Electronic Industries Association RS232-C Standard

EIA RS232-C is a standard for interface between data terminal equipment and data communications equipment employing serial binary data interchange. This standard completely defines all aspects of a serial by bit communications protocol. It includes specifications for the electrical characteristics of the interface and interchange circuits,

the signal levels for transmission and reception, the mechanical characteristics of the connection to the interface, the functional description of the interface circuits, and a description of a selected set of data transmission configurations. This standard defines all necessary specifications for performance at the interface between a serial-by-bit communications interface (e.g., a MODEM which interfaces to the telephone system) and a serial-by-bit data terminal (e.g., a computer terminal). This standard was first published in 1969, and has since been expanded by RS422 and RS423, which describe balanced and unbalanced updates for the RS232-C standard. These expanded standards also extend the maximum line length from 50 ft (15 m) to 4000 ft (1200 m) and up the maximum bit per second (baud) rate to 10 megabits/second.

The EIA RS232-C standard has become a Government standard and is required for use in all Government serial interface applications involving automated digital computing equipment, as detailed in the Federal Information Processing Standards (FIPS) publications. The FIPS documents also define the code to be used over the RS232-C communications circuits as standard ASCII (American Standard for Computer Information Interchange) or a subset thereof. As a result, there is a great deal of RS232-C terminal equipment in use, and it is the most prolific serial-by-bit interface standard. Furthermore, LSI technology has produced single integrated circuits for handling serial to parallel and parallel to serial data conversion at the interface, as well as parity generation and checking. Transmit and receive functions are asynchronously implemented on one chip.

The EIA RS232-C standard does not define how the information is coded onto the communication channel. The ASCII standard requires the format shown in Figure 3 for asynchronous data transmission (start-stop). The appropriate seven-bit ASCII character is sent least significant bit (LSB) first, after a start bit is sent. The start bit synchronizes the receiver to the transmitter. After the seven-bit ASCII character is sent, an odd parity bit is inserted; next, two stop bits are sent, returning the line to the quiescent state and ready for another start bit.

This standard can be used in full duplex (simultaneous bidirectional) systems.

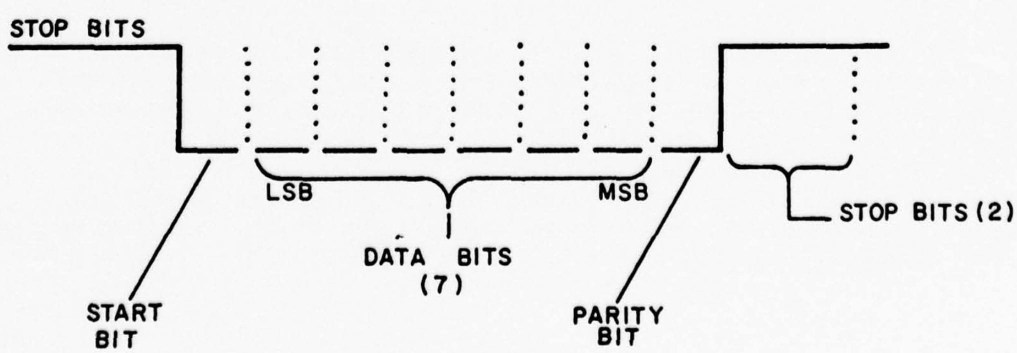


Figure 3. RS232-C/ASCII frame format.

4 PARALLEL DATA COMMUNICATION STANDARDS

Selection of Standards

The selection of candidate parallel interface standards was simplified by considering only those designed for data acquisition and control. Only two standards were found to be appropriate for applications to the proposed EMCS: (1) the IEEE 488 general-purpose interface bus standard (sometimes called HPIB for Hewlett Packard Interface Bus with reference to the company that developed the standard), and (2) the IEEE 583 Modular Instrumentation and Digital Interface System (sometimes called CAMAC, for Computer Automated Measurement and Control).

IEEE-488-1975-Standard

This standard defines the hardware and protocol of a general-purpose digital interface system for limited distance applications (less than 20 m), and allows for simple connections and communications among the digital input/output devices connected to the bus. The bus consists of 16 lines, eight bidirectional data lines, three bidirectional byte transfer control lines, and five bidirectional interface management lines (see Figure 4). Table 1 describes how combinations of signals on the control and management lines define interface operations.

The interface standard provides processor control of all devices on the bus, so that these devices communicate directly by being defined as either "talkers" or "listeners" by the processor.

The operation of the HPIB in state diagrams is described in 33 pages of the IEEE 488 Standard Manual. This complexity is designed into the standard to allow for interfacing as many different transfers on the bus as possible, and to allow for interfacing as many different peripheral devices as possible. A description of a simple data acquisition cycle from an analog multiplier A/D converter peripheral will illustrate this point.

It is first necessary to clear the interface system. To do this, the processor sets IFC (interface clear). Next, the processor must reset the peripheral devices to a known quiescent state by sending a DCL (device clear) on the data bus lines. The processor next sends the listen address of the analog to digital subsystem on the data bus lines. Then the processor sends the channel number of the analog multiplex channel of the data lines; also encoded in this channel address is the start command to the A/D converter. The processor then sends the UNL

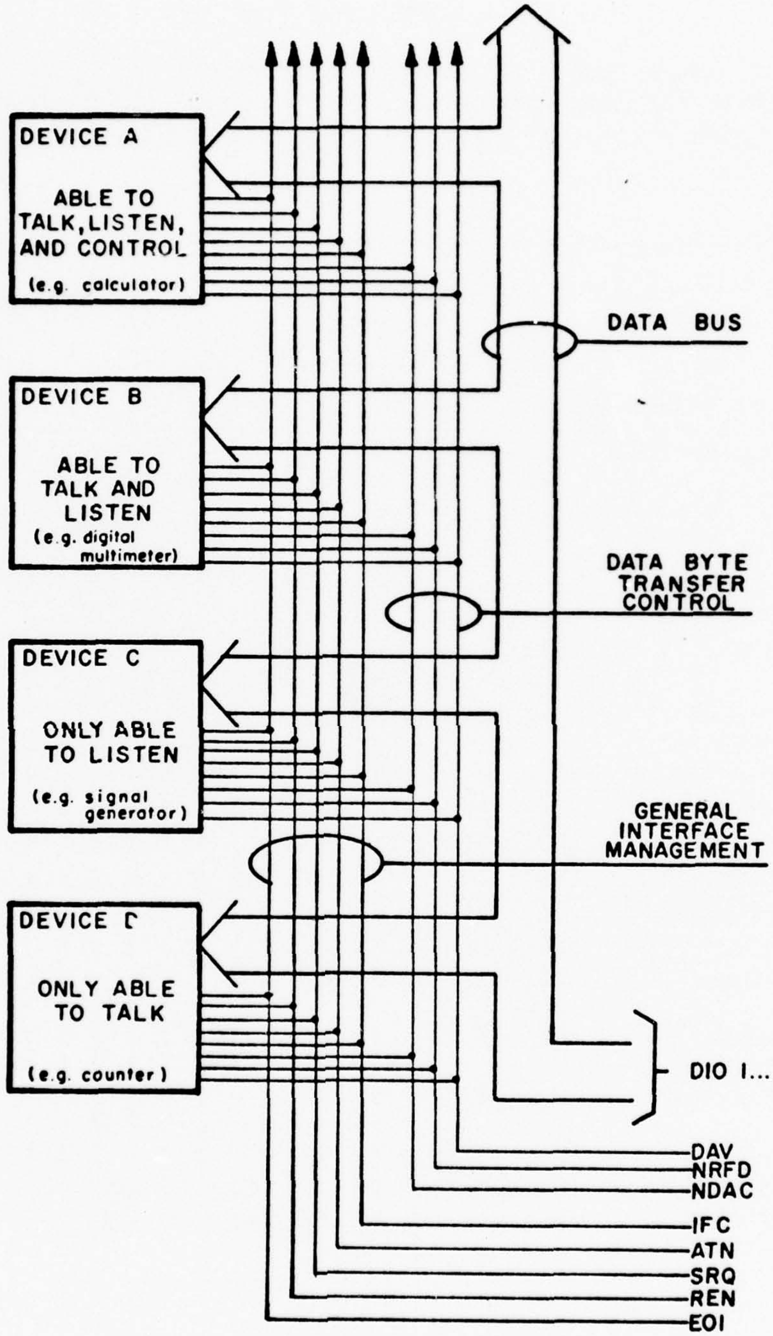


Figure 4. IEEE 488 interface capabilities and bus structure. (From R. A. Donnan and J. R. Kersey, "Synchronous Data Link Control: A Perspective," *IBM Systems Journal*, Vol 13, No. 2 [1974] pp 140-162. Reprinted by permission of International Business Machines Corporation.)

Table 1
IEEE 488 Interface Bus Functions

(From R. A. Donnan and J. R. Kersey, "Synchronous Data Link Control: A Perspective," *IBM Systems Journal*, Vol 13, No. 2 [1974] pp 140-162. Reprinted by permission of International Business Machines Corporation).

					Bus Signal Line(s) and Coding That Asserts the True Value of the Message													
MNEMONIC	MESSAGE NAME		TYPE	CLASS	DIO8	7	6	5	4	3	2	DIO1	DAV NRFD NDAC	ATN	EOI	SRQ	IFC	REN
ACG	addressed command group	(Note 10)	M	AC	X	Ø	Ø	Ø	X	X	X	X	XXX	X	X	X	X	X
ATN	attention		U	UC	X	X	X	X	X	X	X	X	XXX	1	X	X	X	X
DAB	data byte	(Notes 1,9)	M	DD	D8	D7	D6	D5	D4	D3	D2	D1	XXX	X	X	X	X	X
DAC	data accepted		U	HS	X	X	X	X	X	X	X	X	XXØ	X	X	X	X	X
DAV	data valid		U	HS	X	X	X	X	X	X	X	X	1XX	X	X	X	X	X
DCL	device clear	(Note 10)	M	UC	X	Ø	Ø	1	Ø	1	Ø	Ø	XXX	X	X	X	X	X
END	end	(Notes 9,11)	U	ST	X	X	X	X	X	X	X	X	XXX	X	1	X	X	X
EOS	end of string	(Notes 2,9)	M	DD	E8	E7	E6	E5	E4	E3	E2	E1	XXX	X	X	X	X	X
GET	group execute trigger	(Note 10)	M	AC	X	Ø	Ø	Ø	1	Ø	Ø	Ø	XXX	X	X	X	X	X
GTL	go to local	(Note 10)	M	AC	X	Ø	Ø	Ø	Ø	Ø	Ø	1	XXX	X	X	X	X	X
IDY	identify	(Notes 10,11)	U	UC	X	X	X	X	X	X	X	X	XXX	X	1	X	X	X
IFC	interface clear		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	1	X
LAG	listen address group	(Note 10)	M	AD	X	Ø	1	X	X	X	X	X	XXX	X	X	X	X	X
LLO	local lock out	(Note 10)	M	UC	X	Ø	Ø	1	Ø	Ø	Ø	1	XXX	X	X	X	X	X
MLA	my listen address	(Notes 3,10)	M	AD	X	Ø	1	L5	L4	L3	L2	L1	XXX	X	X	X	X	X
MTA	my talk address	(Notes 4,10)	M	AD	X	1	Ø	T5	T4	T3	T2	T1	XXX	X	X	X	X	X
MSA	my secondary address	(Notes 5,10)	M	SE	X	1	1	S5	S4	S3	S2	S1	XXX	X	X	X	X	X
NUL	null byte		M	DD	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	XXX	X	X	X	X	X
OSA	other secondary address	(Note 10)	M	SE	(OSA = SCG $\overline{\text{MSA}}$)													
OTA	other talk address	(Note 10)	M	AD	(OTA = TAG $\overline{\text{MTA}}$)													

Table 1 (cont'd)

MNEMONIC	MESSAGE NAME		TYPE	CLASS	DIO 8	7	6	5	4	3	2	DIO 1	DAV NRFD NDAC	ATN	EOI	SRQ	IFC	REN
					Bus Signal Line(s) and Coding That Asserts the True Value of the Message													
PCG	primary command group	(Note 10)	M	--	(PCG = ACG v								UCG v LAG v TAG)					
PPC	parallel poll configure	(Note 10)	M	AC	X	Ø	Ø	Ø	Ø	1	Ø	1	XXX	X	X	X	X	X
PPE	parallel poll enable	(Notes 6,10)	M	SE	X	1	1	Ø	S	P3	P2	P1	XXX	X	X	X	X	X
PPD	parallel poll disable	(Notes 7,10)	M	SE	X	1	1	1	D4	D3	D2	D1	XXX	X	X	X	X	X
PPR1	parallel poll response 1		U	ST	X	X	X	X	X	X	X	1	XXX	X	X	X	X	X
PPR2	parallel poll response 2		U	ST	X	X	X	X	X	X	1	X	XXX	X	X	X	X	X
PPR3	parallel poll response 3		U	ST	X	X	X	X	X	1	X	X	XXX	X	X	X	X	X
PPR4	parallel poll response 4		U	ST	X	X	X	X	1	X	X	X	XXX	X	X	X	X	X
PPR5	parallel poll response 5		U	ST	X	X	X	1	X	X	X	X	XXX	X	X	X	X	X
PPR6	parallel poll response 6		U	ST	X	X	1	X	X	X	X	X	XXX	X	X	X	X	X
PPR7	parallel poll response 7		U	ST	X	1	X	X	X	X	X	X	XXX	X	X	X	X	X
PPR8	parallel poll response 8		U	ST	1	X	X	X	X	X	X	X	XXX	X	X	X	X	X
PPU	parallel poll unconfigure	(Note 10)	M	UC	X	Ø	Ø	1	Ø	1	Ø	1	XXX	X	X	X	X	X
REN	remote enable		U	UC	X	X	X	X	X	X	X	X	XXX	X	X	X	X	1
RFD	ready for data		U	HS	X	X	X	X	X	X	X	X	XØX	X	X	X	X	X
RQS	request service	(Note 9)	U	ST	X	1	X	X	X	X	X	X	XXX	X	X	X	X	X
SCG	secondary command group	(Note 10)	M	SE	X	1	1	X	X	X	X	X	XXX	X	X	X	X	X
SDC	selected device clear	(Note 10)	M	AC	X	Ø	Ø	Ø	Ø	1	Ø	Ø	XXX	X	X	X	X	X
SPD	serial poll disable	(Note 10)	M	UC	X	Ø	Ø	1	1	Ø	Ø	1	XXX	X	X	X	X	X
SPE	serial poll enable	(Note 10)	M	UC	X	Ø	Ø	1	1	Ø	Ø	Ø	XXX	X	X	X	X	X
SRQ	service request		U	ST	X	X	X	X	X	X	X	X	XXX	X	X	1	X	X
STB	status byte	(Notes 8,9)	M	ST	S8	X	S6	S5	S4	S3	S2	S1	XXX	X	X	X	X	X
TCT	take control	(Note 10)	M	AC	X	Ø	Ø	Ø	1	Ø	Ø	1	XXX	X	X	X	X	X
TAG	talk address group	(Note 10)	M	AD	X	1	Ø	X	X	X	X	X	XXX	X	X	X	X	X
UCG	universal command group	(Note 10)	M	UC	X	Ø	Ø	1	X	X	X	X	XXX	X	X	X	X	X
UNL	unlisten	(Note 10)	M	AD	X	Ø	1	1	1	1	1	1	XXX	X	X	X	X	X
UNT	untalk	(Note 10)	M	AD	X	1	Ø	1	1	1	1	1	XXX	X	X	X	X	X

Table 2

IEEE 583 Bus Lines

Title	Designation	Contacts	Use at a Module
Command			
Station Number	N	1	Selects the module (individual line from control station)
Subaddress	A1, 2, 4, 8	4	Selects a section of the module
Function	F1, 2, 4, 8, 16	5	Defines the function to be performed in the module
Timing			
Strobe 1	S1	1	Controls first phase of operation (Dataway signals must not change)
Strobe 2	S2	1	Controls second phase (Dataway signals may change)
Data			
Write	W1-W24	24	Bring information to the module
Read	R1-R24	24	Take information from the module
Status			
Look-at-Me	L	1	Indicates request for service (individual line to control station)
Busy	B	1	Indicates that a Dataway operation is in progress
Response	Q	1	Indicates status of feature selected by command
Command Accepted	X	1	Indicates that module is able to perform action required by the command
Common Controls			*Operate on all features connected to them, no command required*
Initialize	Z	1	Sets module to a defined state (Accompanied by S2 and B)
Inhibit	I	1	Disables features for duration of signal
Clear	C	1	Clears registers (Accompanied by S2 and B)
Nonstandard Connections			
Free bus lines	P1, P2	2	For unspecified uses
Patch contacts	P3 - P5	3	For unspecified interconnections. No Dataway lines
Mandatory Power Lines			*The crate is wired for mandatory and additional lines*
+24 V dc	+24	1	
+ 6 V dc	+ 6	1	
- 6 V dc	- 6	1	
-24 V dc	-24	1	
0 V	0	2	Power return
Additional Power Lines			*Lines are reserved for the following power supplies:*
+200 V dc	+200	1	Low current for indicators, etc.
+ 12 V dc	+ 12	1	
- 12 V dc	- 12	1	
117 V ac (Live)	ACL	1	
117 V ac (Neutral)	ACN	1	
Clean Earth	E	1	Reference for circuits requiring clean earth
Reserved	Y1, Y2	2	Reserved for future allocation
Total		86	

(unlisten) command, to clear the data bus. The processor next addresses itself to listen and then sends the talk address of the A/D converter. On completion of the A/D conversion, the converter sends the requested data to the processor on the data lines.

This data input cycle requires eight distinct input/output (I/O) port operations by the processor. It also requires that the peripheral device (in the example, an A/D converter subsystem) have sufficient electronics to recognize the different set patterns and states of the bus, and respond properly.

IEEE-583-1975 Standard

This standard fully specifies a data bus by means of which instruments and other functional modules can communicate with each other, with peripherals, and with processors. This standard reduces both the variety and quantity of interfacing required in a single installation and provides a considerable degree of processor independence. All necessary peripheral interfacing is typically supplied via a single processor-dependent interface (dataway controllor) between the processor and the interface system bus controller, as shown in Figure 5.

Table 2 summarizes the signal lines of the dataway. The following constitute a command: the state of the signals on the individual station number lines (specifying a module), the four subaddress lines (specifying a section of the module that would be occupied by a data input channel or control function output), and the five function lines (specifying the type of operation). This yields a maximum of 16 operating channels within a station; 32 different function codes are possible for each channel.

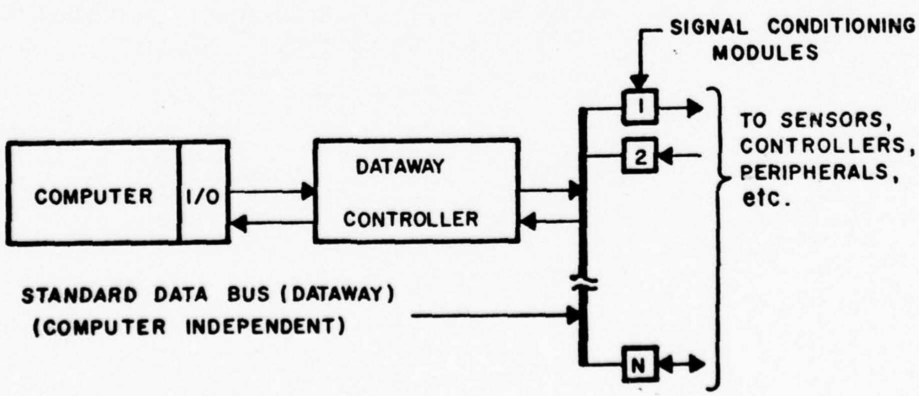


Figure 5. Dataway controller interfacing.

Table 3 summarizes the definitions of the 32 function codes. The standard function codes cause specific operations to be performed by the electronics package that occupies the slot in the selected module defined by the four subaddress lines. Some function codes are reserved for future standard definition, and some function codes remain non-standard for individual custom function definition. Two registers are defined within the function codes, thus allowing each channel to have a data register for inputting data or storing control setpoints, and a status register to report on the status of the electronics package occupying that subaddress.

The function codes provide for all necessary operations to be performed on the individual bits of both registers in each subaddress location. They also provide for enabling and inhibiting functions and for polling the subaddress locations to test for service requirements.

An example of a typical implementation of the IEEE-583-1975 interfacing standard is the system shown in Figure 5. This system is purposely configured to demonstrate the implementation of the standard in the FID. Requests for data input from sensors are performed by writing the selected data channel number into the group 2 register, using function code F(17), overwrite group 2 register (Table 3). The building controller processor then continuously tests the "Look-at-Me" function by using function code F(8) (Table 3). When the data requested is available, the building controller processor requests the read group 1 register, function code F(0) (Table 3). This causes the data to be transferred to the building controller. Note that all commands are given to the dataway controller, which in turn handles the actual dataway.

The return of a control setpoint is accomplished similarly. The building controller presents the control setpoint on the data lines and uses function code F(16) (Table 3) in conjunction with the appropriate subaddress of the correct controller. The dataway controller then deposits this control setpoint into the data (output) register 1 selected by the subaddress.

Table 3

IEEE 583 Function Codes

Code F( )	Function	Use of R and W Lines	Function Signals F16	F8	F4	F2	F1	Code F( )
0	Read Group 1 register	Functions using the R lines	0	0	0	0	0	0
1	Read Group 2 register		0	0	0	0	1	1
2	Read and Clear Group 1 register		0	0	0	1	0	2
3	Read Complement of Group 1 register		0	0	0	1	1	3
4	Nonstandard		0	0	1	0	0	4
5	Reserved		0	0	1	0	1	5
6	Nonstandard		0	0	1	1	0	6
7	Reserved		0	0	1	1	1	7
8	Test Look-at-Me	Functions using the R or W lines	0	1	0	0	0	8
9	Clear Group 1 register		0	1	0	0	1	9
10	Clear Look-at-Me		0	1	0	1	0	10
11	Clear Group 2 register		0	1	0	1	1	11
12	Nonstandard		0	1	1	0	0	12
13	Reserved		0	1	1	0	1	13
14	Nonstandard		0	1	1	1	0	14
15	Reserved		0	1	1	1	1	15
16	Overwrite Group 1 register	Functions using the W lines	1	0	0	0	0	16
17	Overwrite Group 2 register		1	0	0	0	1	17
18	Selective Set Group 1 register		1	0	0	1	0	18
19	Selective Set Group 2 register		1	0	0	1	1	19
20	Nonstandard		1	0	1	0	0	20
21	Selective Clear Group 1 register		1	0	1	0	1	21
22	Nonstandard		1	0	1	1	0	22
23	Selective Clear Group 2 register		1	0	1	1	1	23
24	Disable	Functions not using the R or W lines	1	1	0	0	0	24
25	Execute		1	1	0	0	1	25
26	Enable		1	1	0	1	0	26
27	Test Status		1	1	0	1	1	27
28	Nonstandard		1	1	1	0	0	28
29	Reserved		1	1	1	0	1	29
30	Nonstandard		1	1	1	1	0	30
31	Reserved		1	1	1	1	1	31

5 EVALUATION OF INTERFACE STANDARDS

Parallel Interface Standards

The IEEE 488 interface standard (HPIB) implements the required functions on a single 16-line bus structure, which is very desirable; however, the hardware and software overhead required to implement this bus standard is excessive, because of the implementation of capabilities not required for the EMCS application. This is true even though the implementation of the hardware required for the IEEE-488 bus has recently been integrated into one LSI circuit by Motorola; therefore, it is still necessary to provide electronic interfacing with this circuit.

The IEEE-583 interface standard (CAMAC) implements the required input/output and status functions with a small amount of hardware and software overhead; however, as defined, the CAMAC bus is 86 lines wide. This is excessive for the EMCS application, in which data input and output do not exceed a width of 8 bits.

Serial Interface Standards

The IBM SDLC standard is implemented on a single LSI circuit recently introduced by Signetics. This standard provides efficient serial data communications that can be made transparent to sender and receiver. Provisions are made within the standard for inserting block checksum characters in the transmitted message to provide automatic error detection. However, the SDLC standard does not provide full duplex operation; responses are made on the basis of a polling scheme by the data link controller.

The RS232-C standard, as updated by the RS422 and RS423 publications, is implemented on a single LSI circuit which is supplied by several manufacturers. The standard provides a method of lateral odd or even parity checking, and both generation and checking are implemented on the LSI circuit. It is possible to provide appropriate checksum block check characters as a function of the communicating devices. The standard adapts readily to full duplex operation because of its asynchronous nature. Furthermore, this standard is the most prolific serial data communications standard in terminal to processor application, especially when the coding scheme is ASCII.

6 CONCLUSIONS

Parallel Interface Standards

The IEEE 488 standard is unacceptable because of the excessive amount of hardware and software needed to support interface functions not required in the EMCS application.

The IEEE 583 standard is acceptable except for the excessive bulk of the standard bus system. This drawback could be overcome by using a scaled-down version which could still meet the established criteria.

Serial Interface Standards

The IBM SDLC standard is unacceptable because of the requirement for polling to accomplish fault communication from the remote building controllers.

The RS232-C standard, as updated by the RS422 and RS423 standards, meets all serial data communications requirements for the proposed EMCS.

7 RECOMMENDATIONS

It is recommended that the following standards be used in the communication lines in the Army standard, nonproprietary EMCS:

1. For the Parallel Data Interface Standard, a modified version of IEEE 583 as shown in the Appendix should be used.

2. For the Serial Data Interface Standard, the RS232-C updated by RS422 and RS423 should be used.

APPENDIX: PARALLEL DATA BASE RECOMMENDATIONS

The IEEE 583 and IEEE 488 parallel interface standards were found to be too bulky or too difficult to implement for the trivial input/output requirements of an FID.

The optimum parallel interface system at the FID should employ a compact internal bus structure as outlined in the IEEE 488 standard, and have the ease of operation as outlined in the IEEE 583 standard. This can easily be achieved by reducing the size of the parallel data input/output buses of the IEEE 583 standard from 24 to 8. Further reduction can be effected by decreasing the number of function lines from five to three, reducing the number of subaddress lines from four to two, and eliminating the power and undefined bus lines. The remaining lines can be placed on a 20-pin connector to the crate.* (See Table A1.)

Table A1

Data and Control Lines Dataway Controller

#	NAME	FUNCTION
1	GND	Digital ground
2	INTR or $\overline{INTR}$	Interrupt to processor
3	SS0	Slot address select 0
4	SS1	Slot address select 1
5	SS2	Slot address select 2
6	CS0	Crate address select 0
7	CS1	Crate address select 1
8	FCN0	Slot function line 0
9	FCN1	Slot function line 1
10	FCN2	Slot function line 2
11	$R/\overline{W}$ or $R/\overline{W}$	Processor data port control
12	DWRP or $\overline{DWRP}$	Data write pulse
13	D0	Data LSB
14	D1	
15	D2	
16	D3	Bidirectional
17	D4	Data Bus
18	D5	
19	D6	
20	D7	Data MSB

*See IEEE Standard 583 for definition of crate.

No attempt is made to define the functions of the eight input and eight output lines; these may be used in any manner by the system designer. Only 11 lines are functionally defined. These lines are used to select the correct slot (subaddress) of the crate (3), to define the function to be performed on that slot (3), to store data into the output register of that slot (1), to select a slot to request service via an interrupt (1), to enable the crate (2), and to direct either read or write control to the processor data I/O port (1). Figures A1 and A2 are proposed schematics to provide proper decoding of these lines. The crate internal bus has only 22 data lines, as shown in Table A2.

Table A2

Crate Internal Bus Structure

NAME	BUS PIN #	FUNCTION
DIGITAL PWR	1	+8UDC to Slots
DIGITAL GND	2	Digital Ground
DB0	3	Data Bus LSB
DB1	4	
DB2	5	
DB3	6	Bidirectional
DB4	7	3-State Crate
DB5	8	Data Bus
DB6	9	
DB7	10	Crate Data Bus MSB
FCN0	11	Slot Function Line 0
FCN1	12	Slot Function Line 1
FCN2	13	Slot Function Line 2
$\overline{\text{DWRP}}$	14	$\overline{\text{Data Write Pulse}}$
INTR	15	Interrupt to Processor
SS0	16	Slot Address Select 0
SS1	17	Slot Address Select 1
SS2	18	Slot Address Select 2
$\overline{\text{EN}}$	19	$\overline{\text{Crate Enable}}$
+ANLOG PWR	20	+18VDC to Slots
ANALOG GND	21	Analog Ground
-ANALOG PWR	22	-18VDC to Slots

The hardware controlling the I/O bus and internal slots has been designed to allow for structured data acquisition and return of control setpoints. These operations are conducted in two phases:

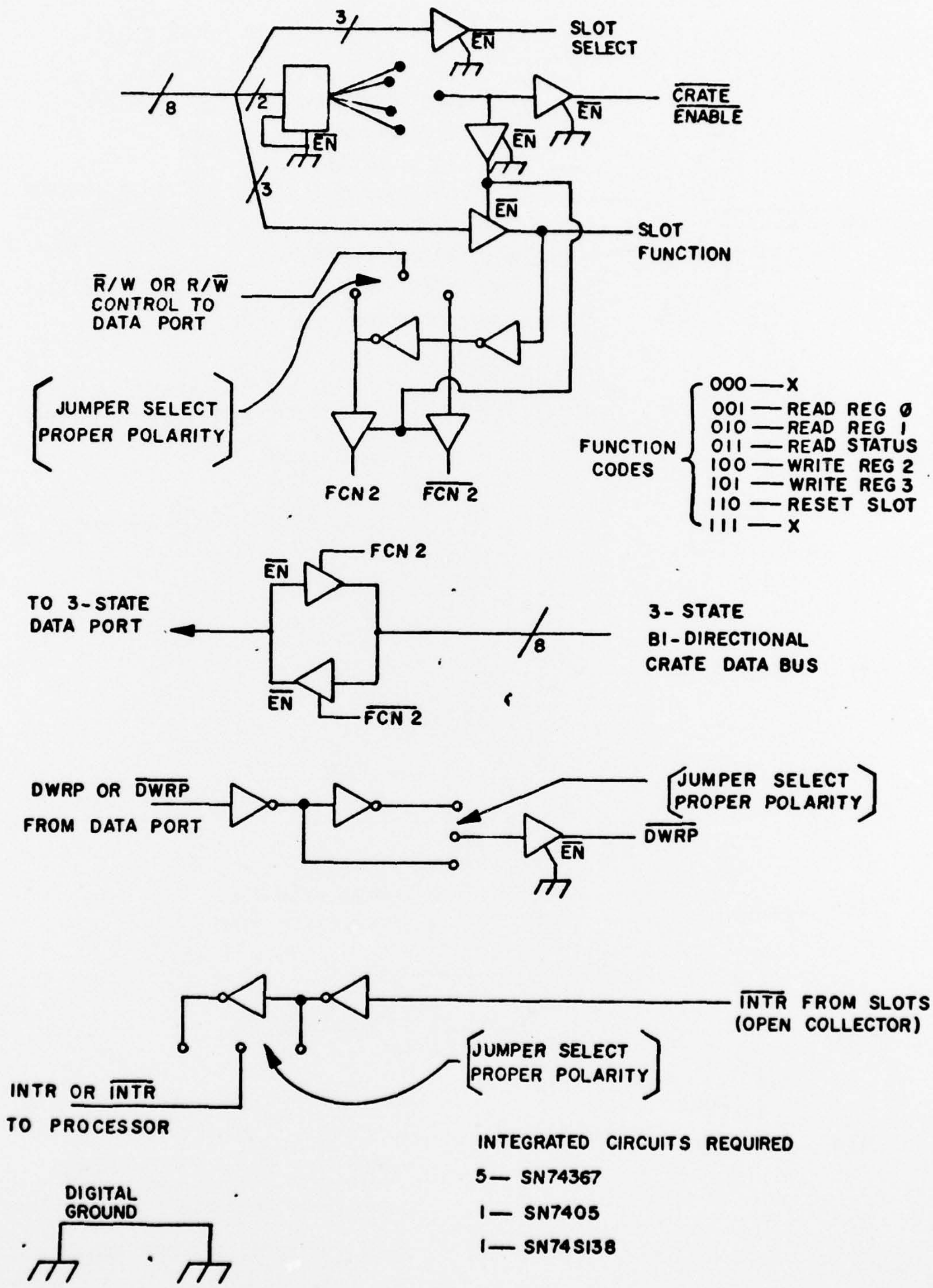


Figure A1. I/O crate bus control schematic.

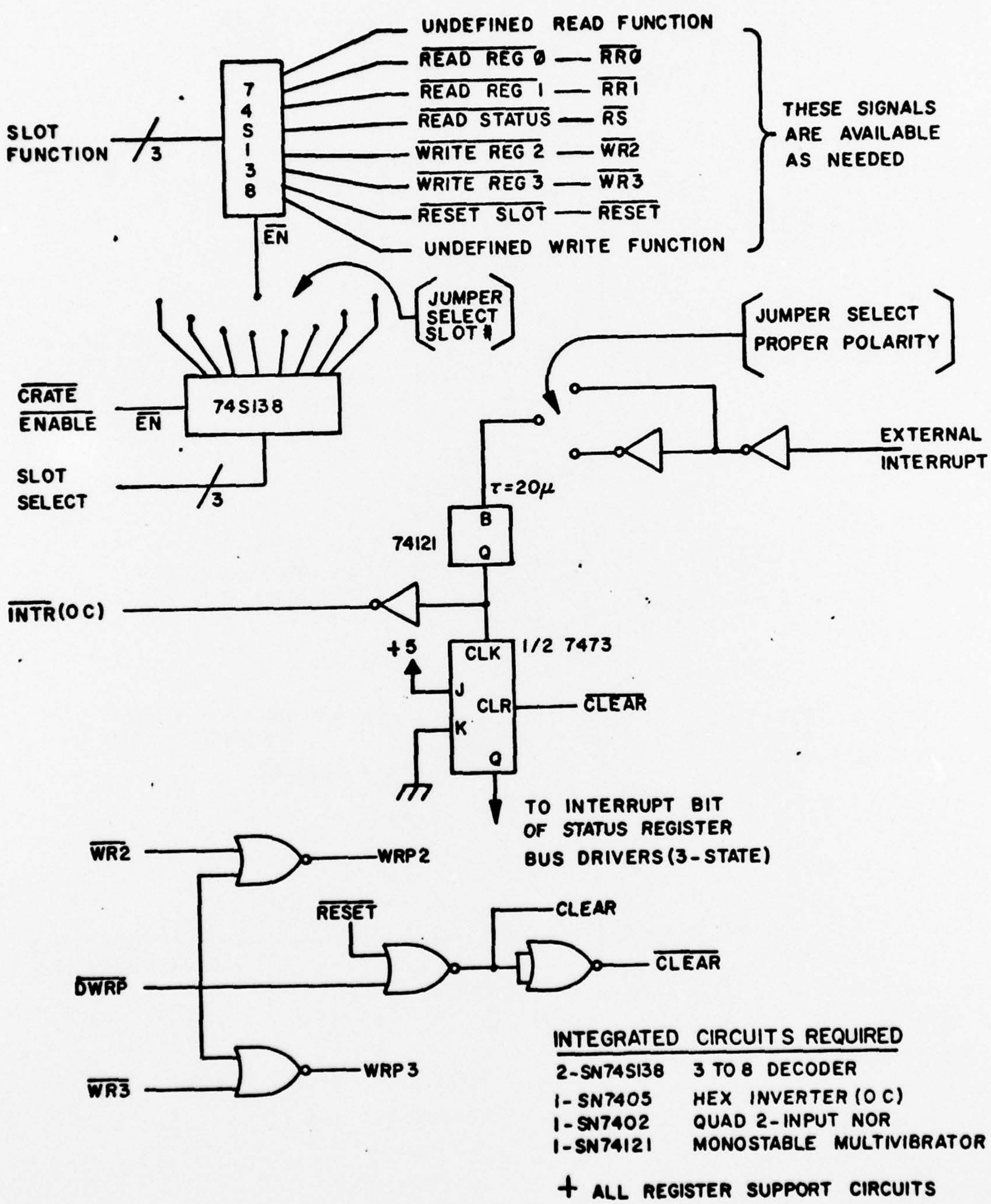


Figure A2. General slot control schematic.

1. Deposit the I/O control byte, which specifies crate number, slot number (of the selected crate), and slot function in the I/O control port. This operation specifies the crate number, slot number of the selected crate, and the function about to be performed with the selected slot. Notice that the crate number and slot number together comprise a five-bit slot selection, allowing for 1 of 32 slots to be selected. Also notice that the slot numbers are sequentially arranged as subsets of the crate numbers, so operations with the slots can easily be performed in incremental fashion. 0, 1, 2, or 3.

2. Perform the required operation (read or write data) at the data port. Sufficient function codes have been provided for easy expansion of up to 16-bit data words. Such higher-accuracy I/O devices must be handled in two passes, since it is necessary to transfer upper- and lower-order bytes to slot registers 0, 1, 2, or 3.

External connection between the processor and the I/O crates is supplied by a 20-pin (or more) "piggyback" connector. This external bus is then supplied in parallel fashion to up to four I/O crates of eight slots each (32 slots total).

The internal crate bus is implemented on a 22 signal bus. This fits well on a standard 22 pin edge connector backplane assembly.

Figures A3, A4, and A5 illustrate various slot control examples.

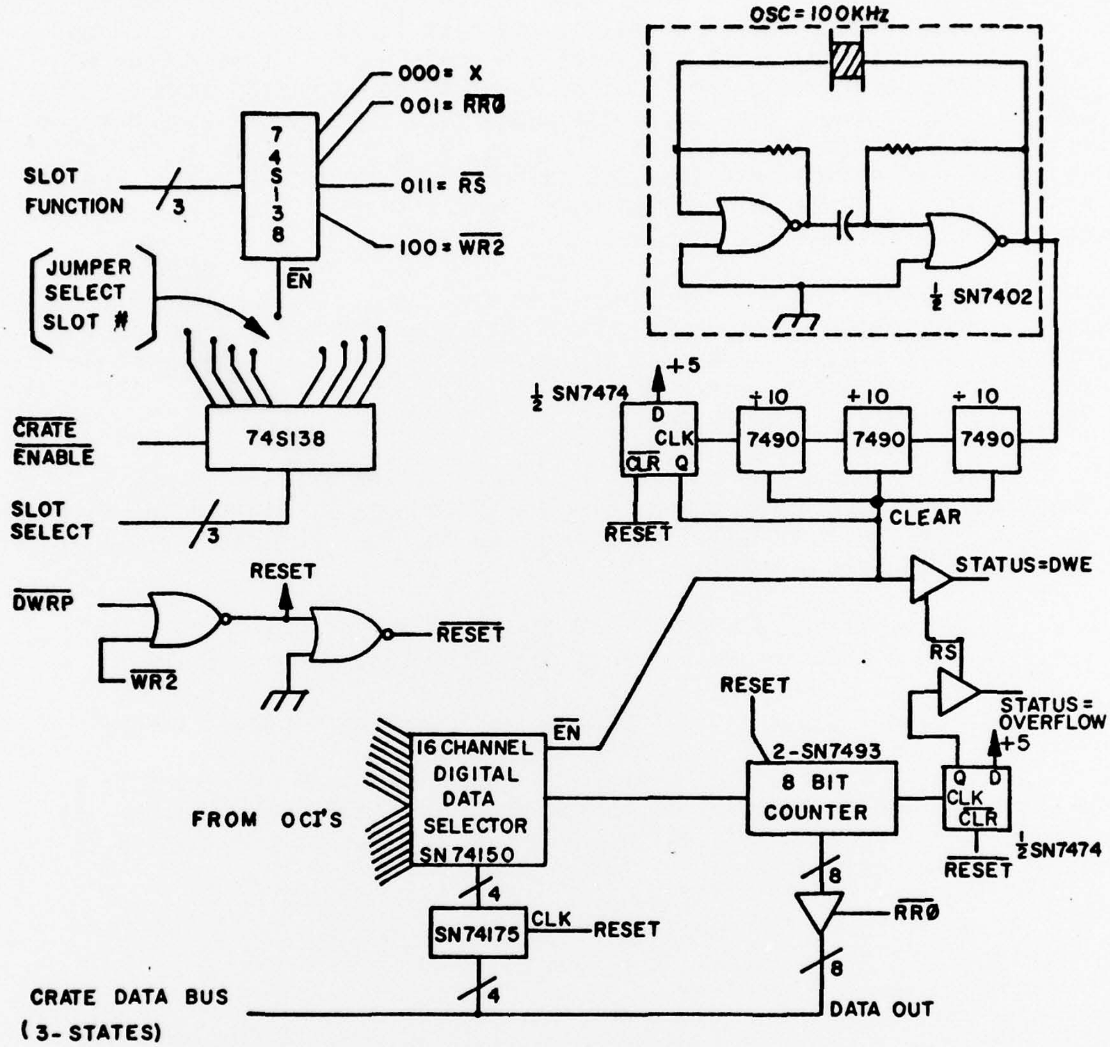
Figure A3 shows an analog function which has been converted to a pulse train whose frequency is representative of the analog value. It is necessary to count this pulse train over a constant time period to determine the analog value.

As configured, time window = 0.01 sec.
Therefore, 100 Hz $\rightarrow$ 1 count
25600 Hz $\longrightarrow$ overflows.

When the data channel number is written into req 2 with $\overline{\text{DWRP}}$ $\overline{\text{WRZ}}$ = reset, the data counter is flash reset for the duration of $\overline{\text{DWRP}}$. Also at this time, the timer counter chain is enabled and done = 0 by reading the status; when done = 1, the data is valid. Overflow is also indicated.

Figure A4 shows a multiplexer to N bit analog to digital (A/D) connector.

Figure A5 shows an eight-channel proportional control return (six-bit accuracy).

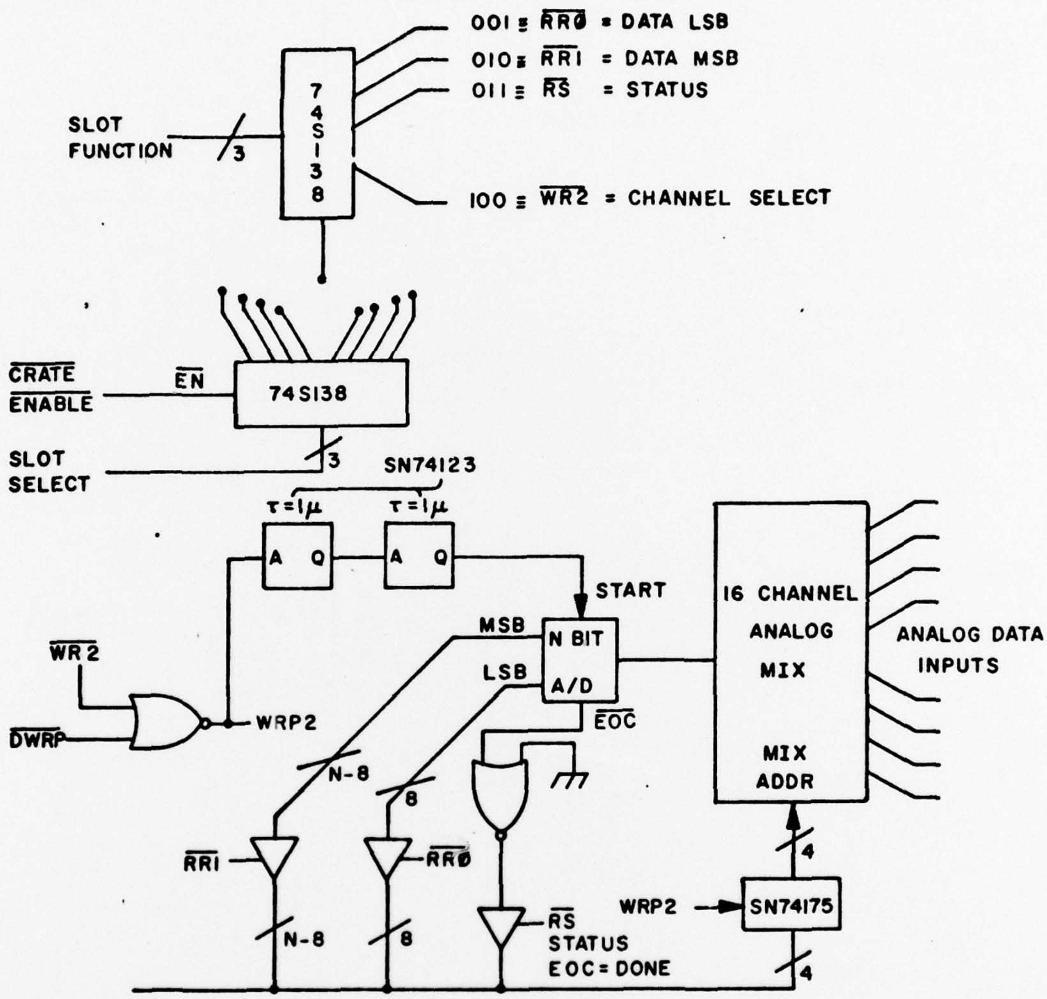


MIX/COUNTER CHIP COMPLEMENT

1—	SN74150	16 TO 1	DATA SELECTOR
2—	SN74S138	3 TO 8	ADDRESS DECODER
1—	SN7474	DUAL	/ TRIGGERED D F F
1—	SN 7402	QUAD	2-INPUT NOR
2—	SN74367	2 & 4	3-STATE BUFFERS
3—	SN7490	DECADE	COUNTERS
2—	SN7493	4 BIT	BINARY COUNTERS
1—	SN74175	4 BIT	LATCH (QUAD DFF)

13 CHIPS

Figure A3. Analog function converted to a pulse train.



CHIP COMPLEMENT

2— SN74S138	3 TO 8 DECODER
1 — SN74123	DUAL MONOSTABLE MULTIVIBRATORS
2 OR 3 SN74367	2&4 3-STATE BUFFERS
1— SN7402	QUAD 2-INPUT NOR
1— SN74175	QUAD DFF
7 OR 8 + ANALOG	MIX & A/D CONVERTER

Figure A4. 16-channel multiplier to N bit A/S converter.

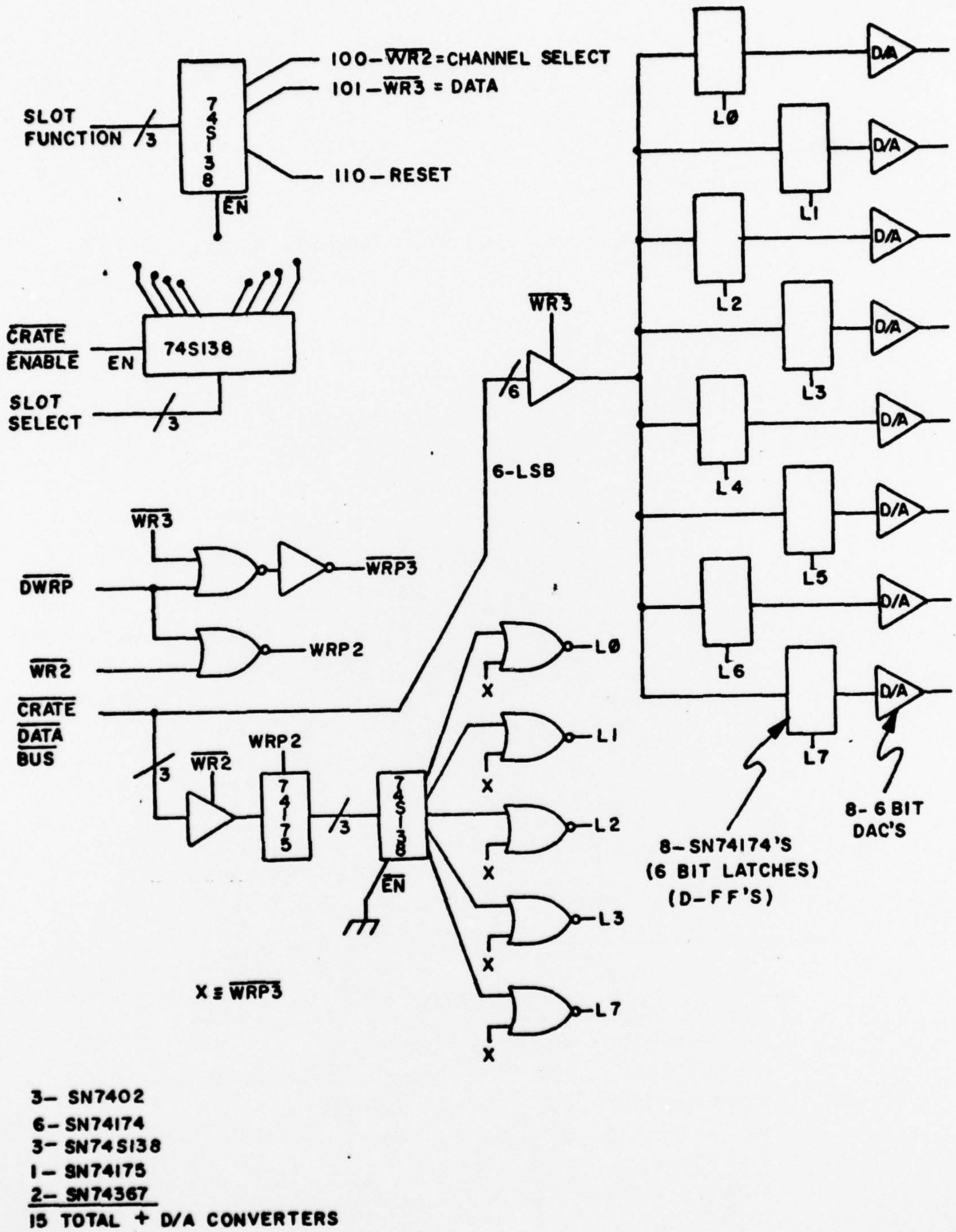


Figure A5. 8-channel proportional control return.

CERL DISTRIBUTION

Chief of Engineers
ATTN: DAEN-ASI-L (2)
ATTN: DAEN-MPO-U
ATTN: DAEN-MPR
ATTN: DAEN-MPE-E
Dept of the Army
WASH DC 20314

Commander
HQDA, TRADOC
ATTN: Engineer Officer
Ft Monroe, VA 23651

Commander
HQDA, DARCOM
ATTN: Engineer Officer
5001 Eisenhower Ave
Alexandria, VA 22314

Commander
HQDA, FORSCOM
ATTN: Engineer Officer
Ft McPherson, GA 30330

US Army Engr Dist, Mobile
ATTN: SAMEN-CI/G. Lickman
P. O. Box 2288
Mobile, AL 36628

Commander
LANTDIVNAVFACENGCOM
ATTN: George Novey/Code 404
Norfolk, VA 23511

Det 1 (CEEDO)
HQ ADTC/ECW/4
ATTN: Michael Mantz
Tyndall AFB, FL 32403

USAF HQ/DEEE
ATTN: Donald Searcy
Langley AFB, VA 23365

Defense Documentation Center
ATTN: DDA (12)
Cameron Station
Alexadria, VA 22314

Commander
HQ, XVIII Airborne Corps and
Fort Bragg
ATTN: AFZA-FE-EE
Fort Bragg, NC 28307

HQ, 7th Army Training Command
ATTN: AETTG-DEH (5)
APO New York 09114

Commander
HQ USAEREUR and 7th Army
ODCS/Engineer
ATTN: AEAEN-EH (4)
APO New York 09403

Commander
7th Army Combined Arms Training
Center
ATTN: AETTM-HRD-EHD
APO New York 09407

US Army Engr Div, Europe
ATTN: Technical Library (3)
APO New York 09757

Commander
V Corps
ATTN: AETVDEH
APO New York 09079

Commander
VII Corps
ATTN: AETSDEH
APO New York 09154

Commander
21st Support Command
ATTN: AEREH
APO New York 09325

Commander
US Army Berlin
ATTN: AEBA-EN
APO New York 09742

Commander
US Army Southern European Task Force
ATTN: AESE-ENG
APO New York 09168

Commander
US Army Installation Support
Activity, Europe
ATTN: AEUES-RP
APO New York 09403

LT Neil B. Hall, CEC, USNR (Code 100)
884-6366
U.S. Navy Public Works Center
Box 6, FPO San Francisco 96651

Hall, James L
Recommended interface standards for an Army standard energy monitoring and control system.--Champaign, IL : Construction Engineering Research Laboratory ; Springfield, VA : available from NTIS, 1978.
33 p. ; 27 cm. (Technical report ; E-140)

1. Buildings--energy consumption. 2. Automatic control. I. Title. II. Title : Army standard energy monitoring and control system. III. Series: U.S. Construction Engineering Research Laboratory. Technical report ; E-140.

DEPARTMENT OF THE ARMY
CONSTRUCTION ENGINEERING RESEARCH LABORATORY
CORPS OF ENGINEERS
P.O. BOX 4005
CHAMPAIGN, ILLINOIS 61820

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD - 314

U.S.MAIL

THIRD CLASS